

Gathering widget traces

This document briefly describes how to gather widget traces that are useful to debug. There are two useful tracing techniques for widgets, one is setting `isDebug` in Dojo to `true`, and the other is setting a trace string for your server.

Turning on widget tracing for both the ClickToCall and CallNotification widgets

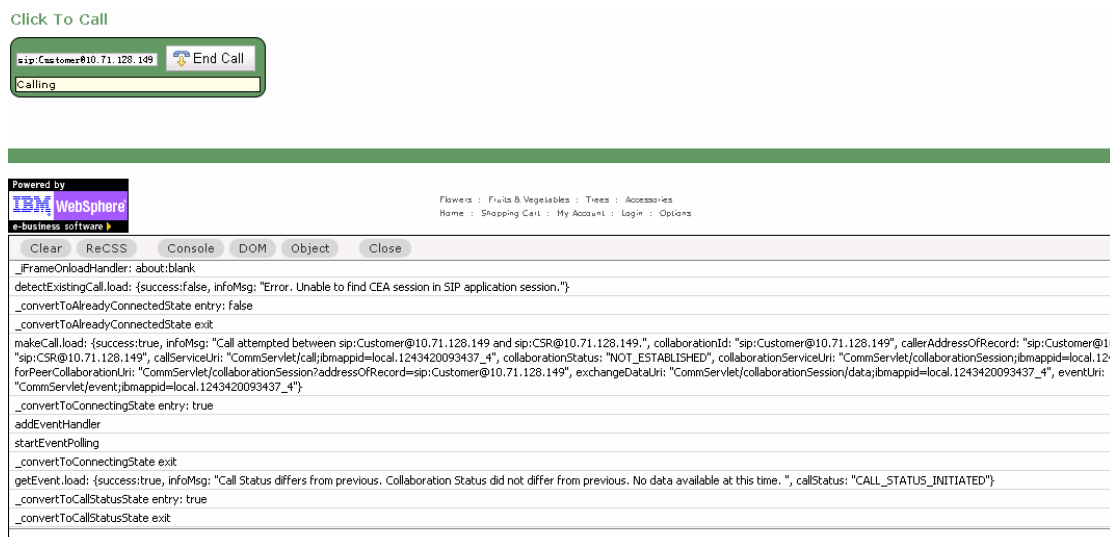
Within your HTML file that contains a CEA widget change the `isDebug` from `false` to `true` in the Dojo include statement.

```
<script type="text/javascript" src="./dojo/dojo.js"
  djConfig=" isDebug: true, parseOnLoad: true">
</script>
```

For example, if you are debugging the Plants by WebSphere sample, you enable the `isDebug` to `true` in the `contactus.html` and `callnotification.html` files.

The details of the trace will show up on Firebug/Firebug Lite when you run the application. You do not need to download any specific product; Firebug Lite is part of the Dojo Toolkit. This "log" is not the server log – it is the output of Firebug Lite. Once you have run through the actions to re-create your error, you will need to copy that output from Firebug Lite to a text file in order to save it.

Example of Firebug Lite output for ClickToCall widget:



The screenshot shows a web browser window with a 'Click To Call' widget. The widget displays 'sip:Customer@10.71.128.149' and an 'End Call' button. Below the widget, the Firebug Lite console is open, showing the following log entries:

```
Powered by IBM WebSphere e-business software
Flowers : Fruits & Vegetables : Tees : Accessories
Home : Shopping Cart : My Account : Login : Options

Clear ReCSS Console DOM Object Close

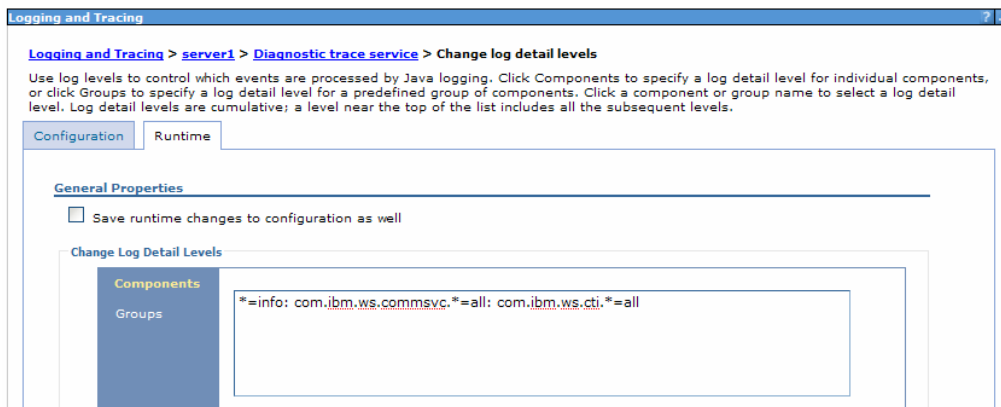
_!FrameOnloadHandler: about:blank
detectExistingCall.load: {success:false, infoMsg: "Error. Unable to find CEA session in SIP application session."}
_convertToAlreadyConnectedState entry: false
_convertToAlreadyConnectedState exit
makeCall.load: {success:true, infoMsg: "Call attempted between sip:Customer@10.71.128.149 and sip:CSR@10.71.128.149.", collaborationId: "sip:Customer@10.71.128.149", callerAddressOfRecord: "sip:Customer@10.71.128.149", callServiceUri: "CommServlet/call;ibmappid=local.1243420093437_4", collaborationStatus: "NOT_ESTABLISHED", collaborationServiceUri: "CommServlet/collaborationSession;ibmappid=local.1243420093437_4", eventUri: "CommServlet/event;ibmappid=local.1243420093437_4"}
_convertToConnectingState entry: true
addEventHandler
startEventPolling
_convertToConnectingState exit
getEvent.load: {success:true, infoMsg: "Call Status differs from previous. Collaboration Status did not differ from previous. No data available at this time.", callStatus: "CALL_STATUS_INITIATED"}
_convertToCallStatusState entry: true
_convertToCallStatusState exit
```

Using the administrative console to enable tracing on a server

For tracing and server logs you will need to turn on the REST and CTI tracing with trace spec: `*=info: com.ibm.ws.commsvc.*=all: com.ibm.ws.cti.*=all`

Procedure

1. Start the administrative console.
2. Click **Troubleshooting > Logs and Trace** in the console navigation tree, then click **server > Diagnostic Trace**.
3. Select the **Runtime** tab.
4. The File Name field specifies the location of where the trace log is located.
5. Select **Change Log Detail Levels**.
6. Select the **Save runtime changes to configuration as well** check box if you want to write your changes back to the server configuration.
7. Change the existing trace state by changing the trace specification to `*=info: com.ibm.ws.commsvc.*=all: com.ibm.ws.cti.*=all`



8. Click **Apply**.